קורס מונחה עצמים – מטלה 3: גרפים ב Python

במטלה זו נממש מבנה נתונים של גרף ממושקל ומכוון ב Python, המימוש כולל מחלקה של גרף וכן מחלקה של אלגוריתמים על גרפים. הרעיון מרכזי הוא להיעזר במימוש הקודם שלכם (של מטלה 2, חלק א') ו"לתרגם" אותו ל Python, ולאחר מכן להשוות את ביצועי הפתרון שלכם למימוש הקודם שלכם ב java, וכן לבצע השוואה למול ספריית NetworkX – יודגש אסור להשתמש בספריות קיימות (בפרט NetworkX), במימוש הפתרון שלכם למטלה (חלק ראשון ושני).

חלק ראשון:

מימוש המחלקה של גרף:

השתמשו בהגדרה של GraphInterface כדי לממש את המחלקה DiGraph.

לאחר שהשלמתם את המחלקה, בדקו את עצמכם בעזרת הפונקציה check0() בקובץ Ex3_main.

לאחר שבדקת שכל השמות נכונים, ממשו מחלקת בדיקה בשם TestDiGraph – שתבצע בדיקה יחידה (UnitTesting) של כל שיטה במחלקה.

חלק שני:

ממשו את המחלקה GraphAlgo שיורשת מהמחלקה (האבסטרקטית) GraphAlgoInterface . בדקו את עצמכם ע"י הרצה של הפונקציות chech1(), check2() בקובץ Ex3_main. לאחר שבדקת שכל השמות נכונים, ממשו מחלקת בדיקה בשם TestGraphAlgo שתבצע בדיקה יחידה (UnitTesting) של כל שיטה במחלקה.

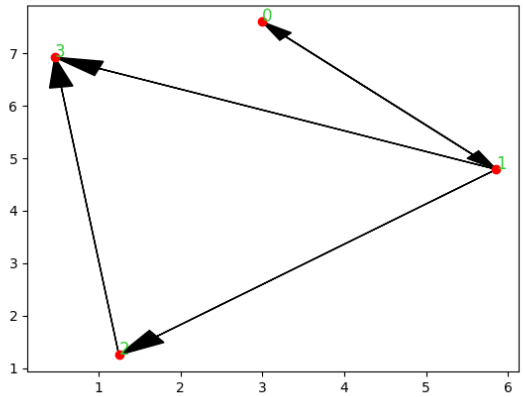
חלק שלישי:

בחלק זה עליכם לבצע השוואה של הקוד שכתבתם למול פתרונות מקבילים, ההשוואה צריכה לכלול את המרכיבים הבאים:

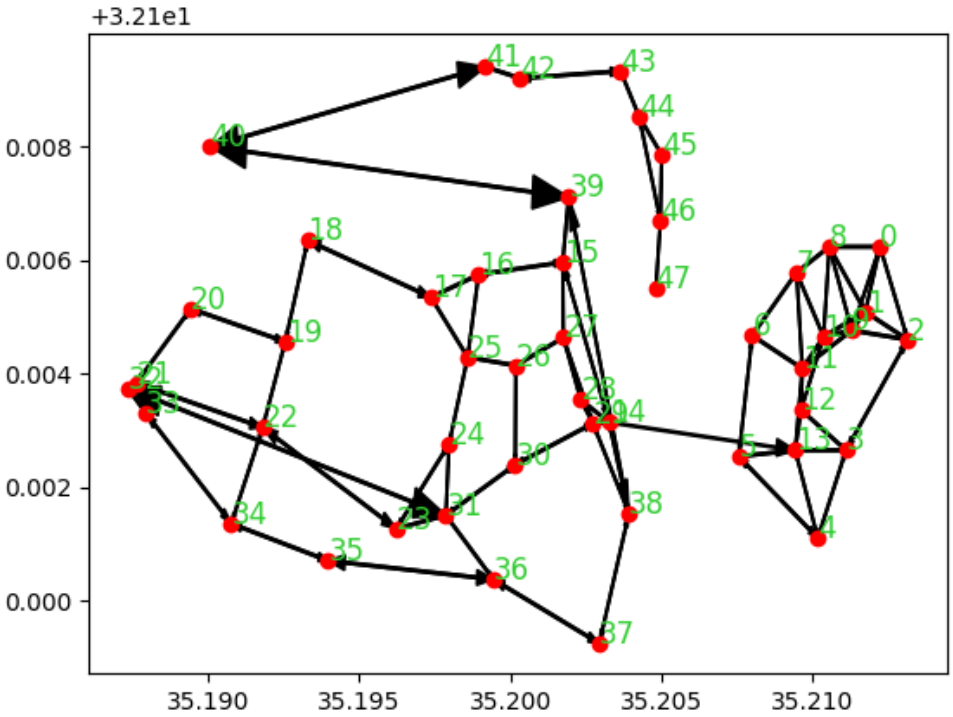
3.1 בדיקת נכונות – על אותם קובצי בדיקה (אותם קבצים של json).

3.2 בדיקה השוואתית של זמן ריצה על אותם תרחישים – בדגש על אלגוריתמים.

- צרו כמה גרפים בגדלים שונים ($100, 10^4, 10^6 \pm$ קודקודים)
- השוואה מול המטלה 2 (חלק ראשון) שלכם, לצורך כך תצטרכו לממש כמה פונקציות נוספות בפתרון מטלה 2 שלכם (ב java).
- השוואה מול NetworkX (הספרייה ב python לגרפים) - גם כאן תצטרכו לממש פתרון שיאפשר הרצה השוואתית של השיטות שנדרשתם לממש במטלה 3, ב NetworkX.
- יש להשוות זמני ריצה, נא לציין מפרט מחשב (3/4/5i, זיכרון ,מספר מעבדים וכו' צבע פחות רלוונטי)
- לצרף גרפים של התוצאות, ניתן להשתמש בmatplotlib או באקסל (לא לצייר על דף ולצלם)
- יש להשוואות את הפונקציות הבאות:
  - shortest_path
  - connected_component
  - connected_components



איור 1: צילום מסך של התוכנית Ex3_main (פונקציה check1) שמציגה גרף עם 4 קודקודים (ללא מיקומים קבועים, אלא כאילו שנבחרו באקראי).



איור 2: צילום מסך של התוכנית Ex3_main (פונקציה check2) שמציגה גרף עם 47 קודקודים (הקובץ A5 מהמטלה הקודמת שכולל מיקומים קבועים – ובהתאם צרכים להיות מוצגים בהתאם).

<u>הנחייה כללית:</u>

- מטלה זו מוגדרת בעיקר ע"י מספר ממשקים שמגדירים את ה api הנדרש ממחלקות, לנוחיותכם מימשנו עבורכם קובץ בדיקת שמות בשם Ex3_main, אנא עשו בו שימוש לבדיקת השמות של המטלות שלכם.
- את הדוח ההשוואתי יש לסכם כמסמך שייכתב כדף wiki בפרויקט ה github שלכם.
- חובה לעשות שימוש ב UnitTesting, לכל מחלקה לוגית שאתם כותבים.
- אין לשנות את הקבצים (המחלקות האבסטרקטיות) שקיבלתם.

- המטלה מיועדת לזוגות – חשוב מאוד ללמוד לעבוד בזוגות ע"ג אותו repository.
- הנחיות הגשה: את המטלה יש להגיש כפרויקט github, וכן לבדוק אותו במערכת בדיקת המטלות לפי ההנחיות מפורטות שהודגמו בתרגול– מטלות שלא תוגשנה לפי ההנחיות לא תזכנה בציון מלא.
- זוהי מטלה רביעית (ואחרונה) בקורס, שכן המטלה החמישית תבוטל בגלל העדר זמן, ולפיכך משקלה הוא של כשל שתי מטלות – התייחסו אליה בהתאם, בפרט הקפידו לתעד היטב את הפתרון שלכם, כתבו Readme מפורט ודפי wiki שכולל הסברים על מבנה הקוד, אופן מימוש האלגוריתמים, כיצד להוריד ולהריץ וכמובן השוואת ביצועים למול NetworkX, והפתרון שלכם למטלה 2 חלק ראשון.

בהצלחה! (מטלה אחרונה ודי).